2020

Semantic Segmentation Using Modified U-Net Architecture for Crack Detection

Michael Sun
South Dakota State University

SEMANTIC SEGMENTATION USING MODIFIED U-NET ARCHITECTURE FOR CRACK DETECTION

BY

MICHAEL SUN

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Computer Science

South Dakota State University

2020

THESIS ACCEPTANCE PAGE

Michael Sun

This thesis is approved as a creditable and independent investigation by a candidate for the master's degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Kwanghee Won
Advisor Date

George Hamer
Department Head Date

Dean, Graduate School Date

This thesis is dedicated to the safety of construction workers.

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to Professor Kwanghee Won for all the support and guidance during my process of completing this thesis. He consistently allowed this paper to be my work but steered me in the right direction. Thank you for kindly providing the concrete crack images. I would also like to thank Processor Sung Shin for his advice and for helping me push myself. I am grateful that I always had both advisors beside me.

I am grateful to my other thesis committee members, Dr. George Hamer and Dr. Cheng Zhang, for their time and feedback for providing direction for my thesis.

Last but not least, I must express my sincere gratitude to my parents, my sister, and my puppy for being always there when I needed you the most. This accomplishment would not have been possible without them. Thank you.

CONTENTS

ABBREVIATIONS vi

ABSTRACT ix

1. INTRODUCTION 1

2. RELATED WORK 4

3. MATERIAL AND METHODS 8

4. IMPLEMENTATION AND RESULTS 15

5. CONCLUSION 22

LITERATURE CITED 24

ABBREVIATIONS

CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
FHT	Fast Haar transform
FFT	Fast Fourier transform
CPU	Central Processing Unit
GPU	Graphics Processing Unit

LIST OF FIGURES

Figure 1. Typical Convolutional Neural Network architecture 4

Figure 2. Visualization of 5x5 filter convolution..... 5

Figure 3. Easy and hard problem in U-Net..... 7

Figure 4. Local context information in CNN..... 8

Figure 5. Proposed network architecture 9

Figure 6. Concatenation of two features..... 12

Figure 7. Pyramid pooling module in proposed architecture..... 13

Figure 8. Examples of our data set..... 16

Figure 9. Example of misclassified crack image of false positive and false negative..... 18

Figure 10. Accuracy of training..... 19

Figure 11. Loss of training..... 20

Figure 12. Implementation results 21

LIST OF TABLES

Table 1 . Detail of proposed architecture 14

Table 2. Data for train, validation, and test 15

Table 3. Definition of each category for comparison 17

Table 4. U-Net confusion matrix 17

Table 5. U-Net with pyramid pooling module confusion matrix 18

Table 6. Proposed method confusion matrix 18

ABSTRACT

SEMANTIC SEGMENTATION USING MODIFIED U-NET ARCHITECTURE FOR CRACK DETECTION

MICHAEL SUN

2020

The visual inspection of a concrete crack is essential to maintaining its good condition during the service life of the bridge. The visual inspection has been done manually by inspectors, but unfortunately, the results are subjective. On the other hand, automated visual inspection approaches are faster and less subjective. Concrete crack is an important deficiency type that is assessed by inspectors. Recently, various Convolutional Neural Networks (CNNs) have become a prominent strategy to spot concrete cracks mechanically. The CNNs outperforms the traditional image processing approaches in accuracy for the high-level recognition task. Of them, U-Net, a CNN based semantic segmentation method, has been one of the most popular in the deep learning because of its excellent performance in open-source crack classification. Although the results of the trained U-Net look good for some dataset, the model still requires further improvement for the set of hard examples of concrete crack that contains the stain, water-spot, and small width crack.

In this paper, we address the challenging problem of accurately detecting a thin concrete crack. We designed a U-Net like structure that has a contracting path and an expansive path to overcome this challenge and compared it to current models, including original U-Net and pyramid pooling module network. The proposed architecture utilizes multiple feature maps in a down-sampling path to obtain a higher pixel-level

segmentation precision. The down-sampled feature is then up-sampled from the output of the pyramid pooling module [13], giving a binary crack and non-crack semantic segmentation. In the experiment, we have collected hard examples and evaluated the approach. Experimental results demonstrate that the proposed network outperforms the U-Net and a pyramid pooling module network in detecting a thin crack in a noisy environment.

1. INTRODUCTION

The advancement of technology has allowed the usage of automated computer vision to identify concrete cracks. A written guide for automatic structural health monitoring (SHM) is also widely used to facilitate visual inspection [19, 20], as well. Together, the computer-vision technique and SHM achieve consistent crack segmentation. Although the automatic inspection process has many benefits, in practice, crack detection is still conducted manually in many places due to complex inspection processes and a shortage of expertized engineers [31-33]. Consequently, automated concrete crack image processing is a highly researched field as an alternative to a highly inefficient and costly manual monitoring system [5-7].

Infrastructure health monitoring is vital for a bridge to maintain its good condition during the service life. An automated bridge inspection technique has several benefits over manual inspection. It is fast and provides more objective measurements of deficiencies, such as cracks and spalled areas. The output database allows bridge engineers to retrieve inspection results in the past years and examine the progression of deteriorations over time. They can prioritize maintenance with limited resources based on objective evaluation results.

Crack is an important deficiency type that is assessed by inspectors. Certain types of cracks, such as material cracks on reinforced concrete, may not be critical, but they can accelerate rusting of rebars. On the other hand, types such as transversal cracks are critical as they can cause bridge failure. Thus, the automated system should be able to

detect a wide range of cracks that have various values of properties such as orientation, location, width, and length.

There have been efforts to develop automated crack detection methods. Some methods include non-machine learning-based approaches and CNN-based approaches. An example of non-machine is an edge detection technique. The edge detection, including Fast Haar transform (FHT), Fast Fourier transform (FFT), Sobel, and Canny, are evaluated carefully in Spencer et al. [20]. Although FHT is significantly more reliable than the other three edge-detection techniques, it is susceptible to misclassifying crack-like noises [35]. As a result, the conventional edge detection approach derives a low efficiency [30]. In comparison, the CNN-based approaches outperform non-machine learning-based approaches because CNN learns important hierarchical features from labeled image data. An example of a CNN-based approach is Deep CNN, which is one of the most successful machine learning techniques in computer vision tasks, including image classification, object detection, and instance segmentation.

One of the benefits of CNNs is their ability to use semantic segmentation for crack detection by classifying every pixel of an image into a crack or a non-crack pixel. Liu et al. [2] employed U-Net, one of the most popular CNNs for the semantic segmentation task. The U-Net based approach outperformed a CNN based approach proposed by Cha et al. [36]. The study compared the accuracy of Cha's CNN versus U-Net. Fifty-seven images were used in training with precision accuracy yield of 90% for U-Net and inapplicable accuracy for Cha's CNN. The result found the U-Net with higher accuracy compared to Cha's CNN. The VGG16 [22, 23] network was also tested, and from 500 training images, the precision accuracy was 82%. Although the VGG16

network showed a similar precision accuracy yield, the U-Net architecture showed better evidence when considering the number of training images required. The three research studies highlight the potential of excellent segmentation performance using U-Net Architecture.

In this paper, we propose an FCN with an encoder and decoder framework based on U-Net. The encoder-decoder characteristic includes the multiscale feature fusion and up-sampling. The U-Net's ingenious idea of using skip connection achieves high accuracy from a relatively small dataset. Thus, we embrace the U-Net's original design of skip connection to find the concrete crack in personally collected images. The contribution encompasses concatenating preceding feature maps before each pooling operation and utilizing a pyramid pooling module from PSPNet [13].

2. RELATED WORK

2.1 CNN and FCN

CNN is a deep learning algorithm in computer vision. It takes in an input image process through hidden layers and output classes. Every CNN network begins with a convolutional layer. The goal of the first convolution layer is to extract low-level features such as edges, color, and gradient orientation. The second layer would then capture higher-level features such as hands or ears. These features maps are captured with the receptive field. For example, Fig. 2 has an input size 3x8x8 array of pixel values and a filter 3x5x5. The output of the convolution generates a feature map 1x4x4 array of numbers. This process repeats multiple times, and the number of convolution varies depending on the application.

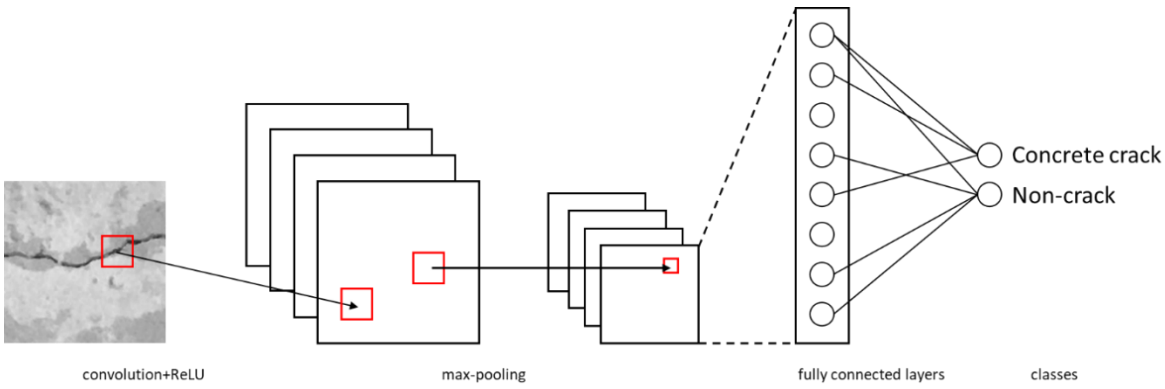


Figure 1. Typical Convolutional Neural Network architecture

The output result of CNN-based object detection, for example, CCRs are indicated by a bounding box [21]. The bounding box only approximates the region of interest, and therefore cannot be used to measure the length, density, or the characteristics

to define the type of crack. It is essential to find pixel-level classification in crack detection for accurate safety analysis.

FCN is derived from a CNN-based segmentation network. It trains end-to-end, pixels-to-pixels digital input images for a given segmentation task. The idea of FCN is to build convolutional layers without any fully connected layers and to produce an output size that corresponds to the input [16]. The input data feature map is encoded and decoded using transposed convolution to attain the same size output. As the network decodes, the skip connection sums pre-extracted feature maps to recover the spatial information during pooling operations.

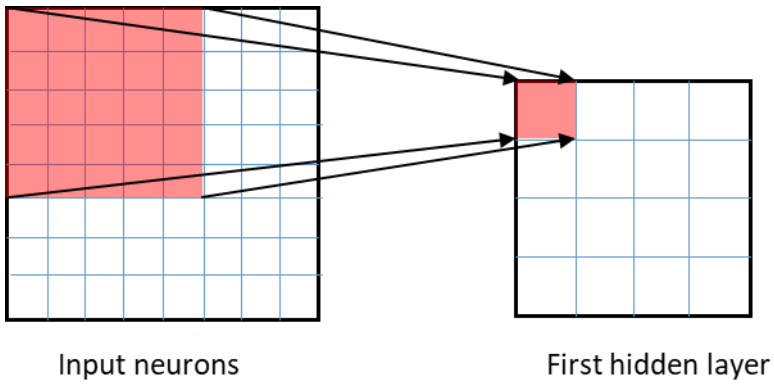


Figure 2. Visualization of 5x5 filter convolution

Examples of FCNs semantic segmentation include ParseNet [12], DeconvolutionNet [15], and U-Net [4]. The ParseNet is an end-to-end convolutional network that predicts the value of every pixel during convolution to keep the global information. In order to do this, the feature maps are reduced and processed with pooling. The context vector is normalized using the L2 Euclidian Norm. ParseNet is able to address the loss of global

context information of the image in its deep layers in FCN [16]. In contrast, DeconvolutionNet utilizes VGG16 architecture. Its input is an instance proposal that is transformed into a vector of features. The features are then de-convoluted using the un-pooling method. The ingenious deconvolution expands feature maps while keeping the information dense to generate pixel-wise classification.

The last segmentation network is U-Net. This network is built around a contracting path and an expansive path of asymmetric u-shape. U-Net is unique in that the expanding part increases the height and width of a reduced number of feature maps into the original size. U-Net does not use any fully connected layer. As a result, it can produce an excellent performance from a relatively small set of training images compared to ParseNet and DeconvolutionNet.

In addition, there are several research studies on the FCNs skip connection. Drozdzal et al. [25] explored the importance of the skip connections where they valued very deep FCN with long and short skip connections. Another study is based on the idea of a dense block proposed by Huang et al. [27]. Zhou et al. [26] utilized the dense block and convolutional layers between the encoder and decoder to strengthen the deep learning. However, the proposed skip pathways and the dense skip connection are unnecessarily elaborate for the crack domain. With this in mind, the parameter of the skip connection should be considered cautiously.

2.2 U-Net

U-Net is an FCN that relies on the use of data augmentation aided toward precise localization in biomedical image segmentation [4]. The U-Net architecture includes

multiple up-sampling layers, skip connection that concatenates feature maps, and learnable weight filters. The result shows outstanding performance in both biomedical image segmentation and crack detection [2].

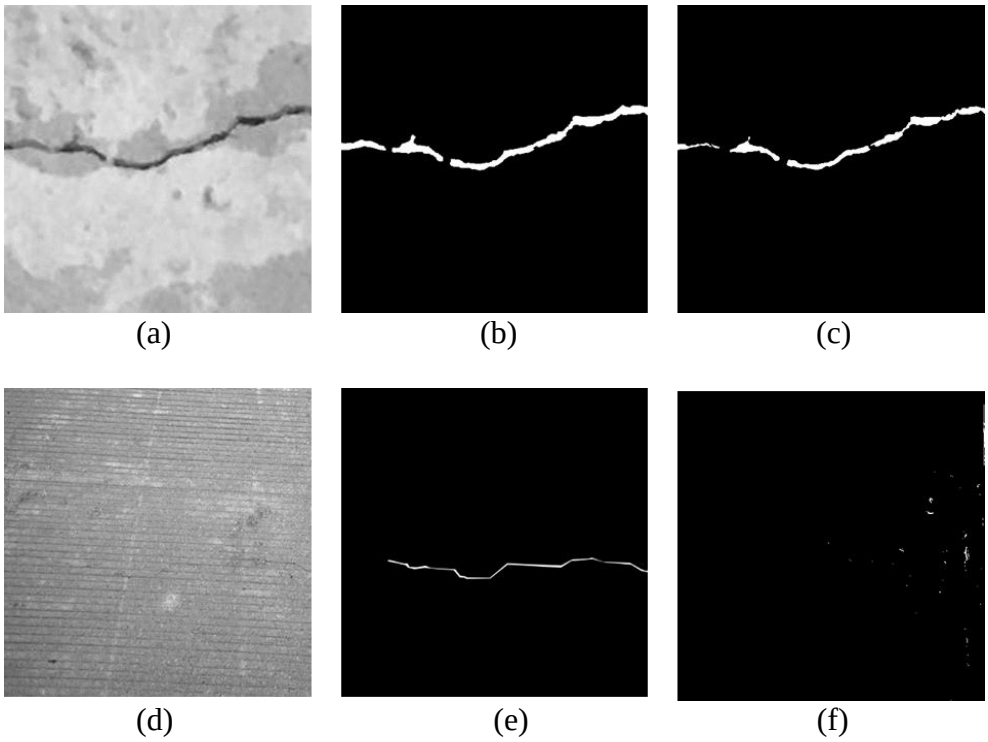
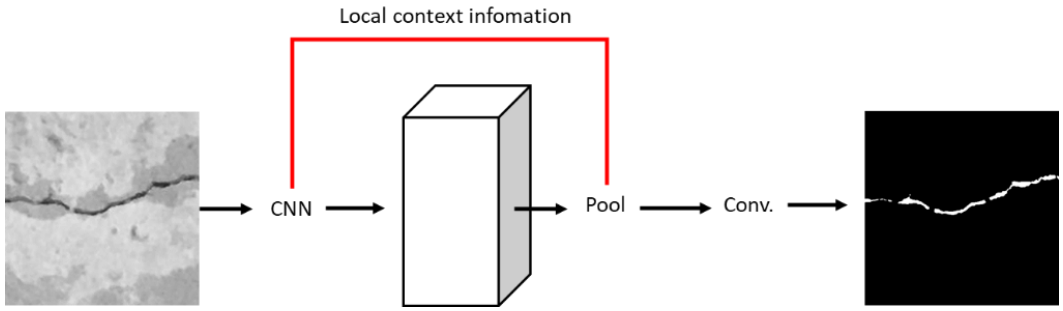


Figure 3. Easy and hard problem in U-Net. (a)(d)original image (b)(e)ground truth (c)(f)output.

2.3 PSPNet

Early algorithms take an input image to CNNs and make feature maps. These feature maps are then up-sampled into final predicted images. During the up-sampling operation, local context information, such as shape and material, are trained. CNNs only use local context information for semantic segmentation.



(a) Input image (b) Feature map (c) Final prediction

Figure 4. Local context information in CNN

Zhao[13] proposed PSPNet to improve the dilated convolution. The network aims to solve common FCN semantic segmentation problems, namely, mismatched relationships, confusion categories, and inconspicuous classes. The proposed pyramid pooling module extracts global context information by different region-based context aggregation and combines it with local context information to produce a better scene parsing task. The pyramid pooling module processes levels of information that differ from global pooling [34]. Additionally, Kim et al. [14] research shows that the pyramid module in U-Net architecture reduces context information loss between different sub-region in high-resolution aerial image segmentation.

3. MATERIAL AND METHODS

The proposed method is based on the U-Net architecture conducive to the evidence of strong performance by a few training data sets in [2, 4, 26]. The proposed method extends U-Net architecture targeting to capture the thin crack in a noisy environment in images.

This section begins with the pipeline of our proposed method for thin concrete segmentation. We first explain the underlying architecture and divide the network into two small groups for a detailed explanation. The proposed network consists of two methods, a concatenation strategy, and the pyramid pooling strategy. The proposed concatenation strategy enables a profound network while the pyramid pooling module facilitates our network to have higher pixel-level classification based on the context information provided.

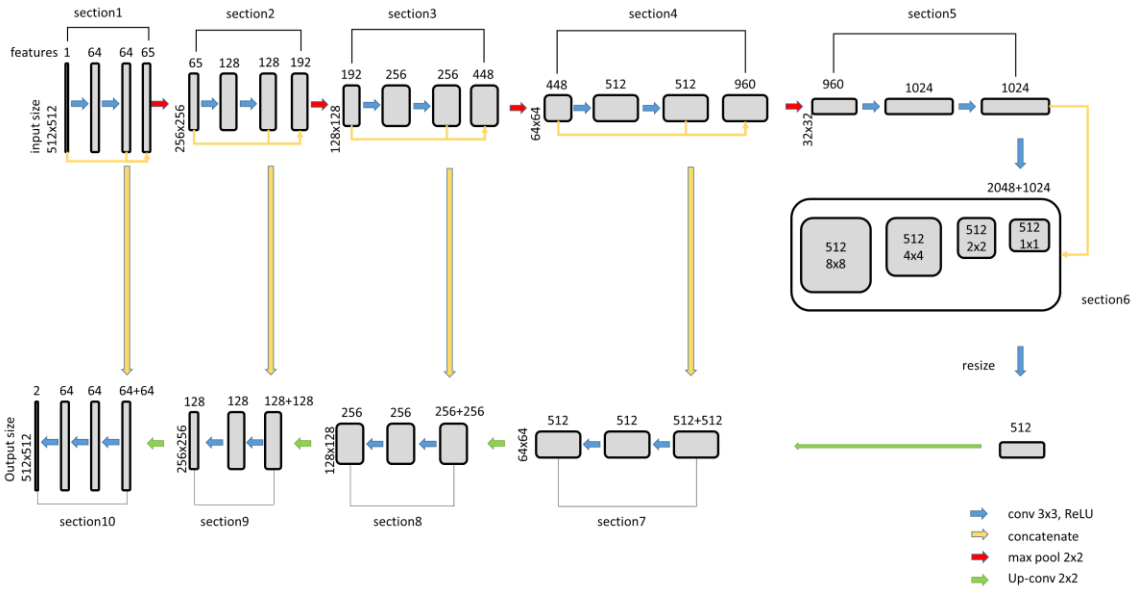


Figure 5. Proposed network architecture

3.1 Observation of U-Net

First, we start with concrete crack failure cases in U-Net architecture. The open-sourced dataset [10] contains concrete crack images represented in Fig. 3(a). The image contains concrete with a color variant of a lighter and darker flat surface that indicates

non-crack. This concrete crack is visible with bare eyes. Our trained U-Net architecture output results in Fig. 3(c) with exceptional performance.

On the other hand, images like Fig. 3 (d) were collected and trained separately from the above open-source dataset. Fig. 3(d) also includes lighter and darker background concrete, but this time the concrete crack is thin, and there are non-crack horizontal lines from the original casting of the surface. This time, the U-Net architecture is unable to detect the crack. It misclassified the water-spots as the crack as seen in Fig. 3(f).

The analysis showcases the need for an improvement in both detailed feature extractions to go deeper into the network to differentiate between the crack and non-crack. This is a challenging problem in concrete crack semantic segmentation that our work aims to improve.

3.2 Architecture

The main framework of the proposed method is illustrated in Fig. 5. It is an end-to-end network with a total of 35 layers. The input images are RGB of 3x512x512 pixels, and the output images are black and white 2x512x512 pixels. The white pixel represents the crack, and the black pixel represents non-crack, as shown in Fig. 3(e). Each convolution uses a 3x3 kernel throughout the network, and convolution is followed by a ReLU function. ReLU has been the most effective among other activation function alternatives as it can train the model much faster along with weight optimization [9]. The activation function is $f(x)=\max(0, x)$ where all negative numbers are set to zero. The max-pooling operation has a 2x2 kernel with stride size 2. We used the pooling kernel size 2x2 since a greater number increases the risk of losing information. The original U-

Net Architecture did not use batch normalization [24]. In ours, each convolution is followed by a batch normalization. This extra step minimizes the overfitting problem.

The first 18 layers are the contracting path. The path divides into five subsections (Fig.5. sections 1-5). Each section consists of two repeated convolutions, a concatenation, and a max-pooling. The pooling doubles the number of feature channels at each down-sampling. The concatenated layer in sections 1-4 will be covered in 3.3 and section 6 in 3.4.

The last 14 layers are the expansive path (sections 7-10). Similar to the contracting path, there are two consecutive convolutions, a concatenation from the skip connection, and an up-convolution. The up-sampled features are concatenated with a corresponding layer from the contracting path repeatedly. The up-convolution doubles the number of pixels and reduces the number of feature channels in half. While the convolution in an expansive path mainly extracts semantic features, up-convolution [11] is used to find the original representation of the convolution matrix. The up-convolution increases the image size, resolving the loss of spatial information in the encoding process.

3.3 Concatenation

This section discusses the operation belonging to sections 1-4 in Fig. 5, as well as the skip connection that concatenates the feature map between the expansive path and the contracting path. We used a series of two 3x3 convolution operations as they resemble a single 5x5 convolution operation [29]. The two consecutive 3x3 operation processes are small compared to the opposing 5x5, and this results in increased computation performance.

Fig. 6 (a) indicates the max-pool feature from a successor layer. The feature map size is then doubled in the next layer (b). The third step concatenates the feature map (a) and (c) together to render (d). For example, section 2 in Fig. 5 concatenates a feature map size 192 (64 and 128). The extra step in (d) enables the network to extract more features and helps to regain some obscure data lost during the convolution and successor max-pooling operation. The effect of (d) accumulates throughout the contraction path until the end of the network. Thus, the accumulation enhances the performance of the deep network when dealing with complex segmentation tasks.

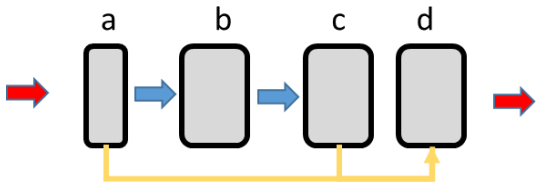


Figure 6. Concatenation of two features. Blue represents convolution, yellow represents concatenation, red represents max-pool

The key idea of FCNs is the skip connection, which is highly valued in our network. The skip connection restores each pixel's precise location by concatenating the output of the transposed convolution layers with the feature maps from the encoder at the same level. Intuitively, the two layers of the same feature map size are combined. The encoder feature map of layer c of Fig. 6 concatenates through the skip connection with the up-convolution layer of the expansive path. Namely, these are 64, 128, 256, and 512 feature maps in Fig. 5.

3.4 Pyramid Pooling Module

Fig. 7 shows the strategy we utilize to use local context information of training crack images from the encoder to extract global context information. In our model, the input of the module is the 32x32 pixels with a 1024 feature map. The 1024 features in Fig. 5 in section 5 process average pooling operation into 1x1 2x2 4x4 8x8 convolution. We employ average-pooling as it has already been proven to be more effective in the pyramid pooling module [13]. The average pooling generates the best outcome of the local and global context information. These are the cracks and non-cracks. We analyzed the two different ratio scales: (1, 2, 4, 8) and (1, 4, 16, 32). The former ratio worked better since the smaller ratio gives global context information. The pyramid module pools 262,144 pixels (512x512) into 1 pixel, 2 pixels, 4 pixels, and 8 pixels, respectively. The four features are then processed through 1x1 convolution and batch normalization. The multiscale feature then concatenates with the last layer of the encoder. The concatenated 3072 (2048+1024) features map is finally reduced into 1/6 in size to match the feature map size of the skip connection. The reduced layer is represented as the last layer in Fig. 7.

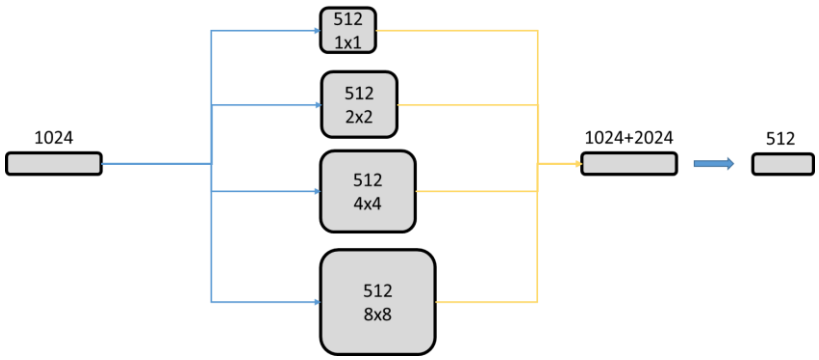


Figure 7. Pyramid pooling module in proposed architecture

Table 1 . Detail of proposed architecture

Steps	Layer	Feature size		Filter
1	Conv	512x512	1	3x3
	Conv	512x512	64	3x3
	Batch_norm	512x512	64	-
	Conv	512x512	64	3x3
	Batch_norm	512x512	64	-
	Concat	512x512	65	-
	Max_pool	256x256	-	2x2
2	Conv	256x256	128	3x3
	Batch_norm	256x256	128	-
	Conv	256x256	128	3x3
	Batch_norm	256x256	128	-
	Concat	256x256	192	-
	Max_pool	128x128	-	2x2
3	Conv	128x128	256	3x3
	Batch_norm	128x128	256	-
	Conv	128x128	256	3x3
	Batch_norm	128x128	256	-
	Concat	128x128	448	-
	Max_pool	64x64	-	2x2
4	Conv	64x64	512	3x3
	Batch_norm	64x64	512	-
	Conv	64x64	512	3x3
	Batch_norm	64x64	512	-
	Concat	64x64	960	-
	Max_pool	32x32	-	2x2
5	Conv	32x32	1024	3x3
	Batch_norm	32x32	1024	-
	Conv	32x32	1024	3x3
	Batch_norm	32x32	1024	-
	Concat	32x32	1024	-
	Drop_out	32x32	-	2x2
6	Avg_pool		1024	2x2
	Conv	1x1, 2x2,	512	3x3
	Batch_norm	4x4, 8x8	512	-
	Concat		3072	-

Steps	Layer	Feature size Filter		Filter
7	Up_sample	64x64	-	-
	Concat	64x64	1024	-
	Conv	64x64	512	3x3
	Batch_norm	64x64	512	-
	Conv	64x64	512	3x3
	Batch_norm	64x64	512	-
8	Up_sample	128x128	-	-
	Concat	128x128	512	-
	Conv	128x128	256	3x3
	Batch_norm	128x128	256	-
	Conv	128x128	256	3x3
	Batch_norm	128x128	256	-
9	Up_sample	256x256	-	-
	Concat	256x256	256	-
	Conv	256x256	128	3x3
	Batch_norm	256x256	128	-
	Conv	256x256	128	3x3
	Batch_norm	256x256	128	-
10	Up_sample	512x512	-	-
	Concat	512x512	128	-
	Conv	512x512	64	3x3
	Batch_norm	512x512	64	-
	Conv	512x512	64	3x3
	Batch_norm	512x512	64	-
11	Conv	512x512	2	3x3
	Batch_norm	512x512	2	-
	Conv	512x512	1	3x3

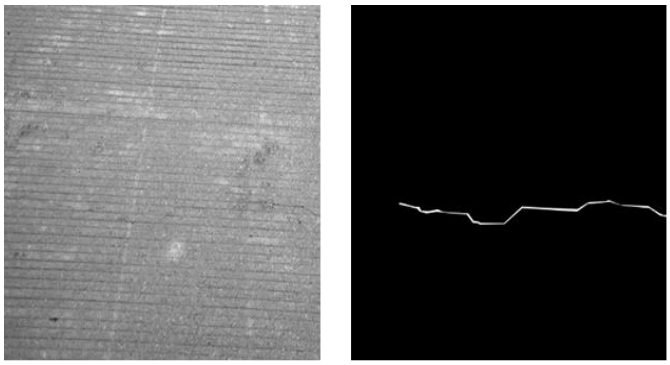
4. IMPLEMENTATION AND RESULTS

4.1 Data Preparation

The dataset has been collected from a concrete bridge located in Lincoln, Nebraska. We have selected 45 images of 2448 by 2448 pixels containing cracks among the images obtained from the top of the bridge deck. The field of view of each image is approximately 2 meters by 2 meters, thus, the size of each pixel is approximately 1 millimeter. Images under different illumination conditions were selected to obtain a robust classification model. As data pre-processing, each image was divided into four sub-sections and cropped into four new images. As a result, we generated 140 images of 512 by 512 pixels. These data were separated into the training, validation, and test sets containing 70, 30, 40 images, respectively, in Table 2. Labeling for each data is a binary image that was hand-generated one by one, Fig. 8(b).

Table 2. Data for train, validation, and test

	No. of images	Size (pixels)	Crack	Non-crack
Train	70	512x512	55	15
Validation	30	512x512	25	5
Test	40	512x512	20	20



(a) (b)

Figure 8. Examples of our data set. (a)image (b)ground truth. In (b), white represents crack, and black represents non-crack.

The data have advantages over the data provided in Ozgenel [10]. Our data contain various adverse factors, including water-spots, shadows, concrete fillers, stains flowing down from the top, and protruding horizontal grids generated from the casts. These non-cracks are often misclassified since they are found to be similar to crack in terms of long and thin geometry and dark color.

4.2 Training and Setup

The model is trained using Google Colab cloud service, Tensorflow 1.14.0., Keras 2.3.1, python3 framework. The hardware specification is as follows: (CPU: Intel(R) Xeon(R) @ 2.30GHz, GPU: Tesla P100-PCIE-16GB). To justify the performance enhancement, we experimented on the same dataset to train and test on the same working environment. The model is trained with 30 epochs with the batch size 32. Binary cross-entropy loss is selected for compilation. For optimizer, we used Adam [28] with a learning rate of 1e-4.

4.3 Analysis

In this section, we evaluate and compare the effectiveness of three models: U-Net, U-Net with pyramid pooling module, and the proposed model. We measured the performance of the model using the confusion matrix (1) because we have a binary classification problem of images with crack and non-crack. The objective of the experiments is to investigate the superiority of the proposed method over the original method U-Net.

Table 3. Definition of each category for comparison

Labels	Meaning
True Positives	predicted crack and the actual output was crack segmentation.
True Negatives	predicted no-crack and the actual output was no-crack segmentation.
False Positives	predicted crack and the actual output was no-crack segmentation.
False Negatives	predicted no-crack and the actual output was crack segmentation.

$$\text{Accuracy} = \frac{\text{TruePositives+FalseNegatives}}{\text{TotalNumberofSamples}} \quad (1)$$

Table 4. U-Net confusion matrix

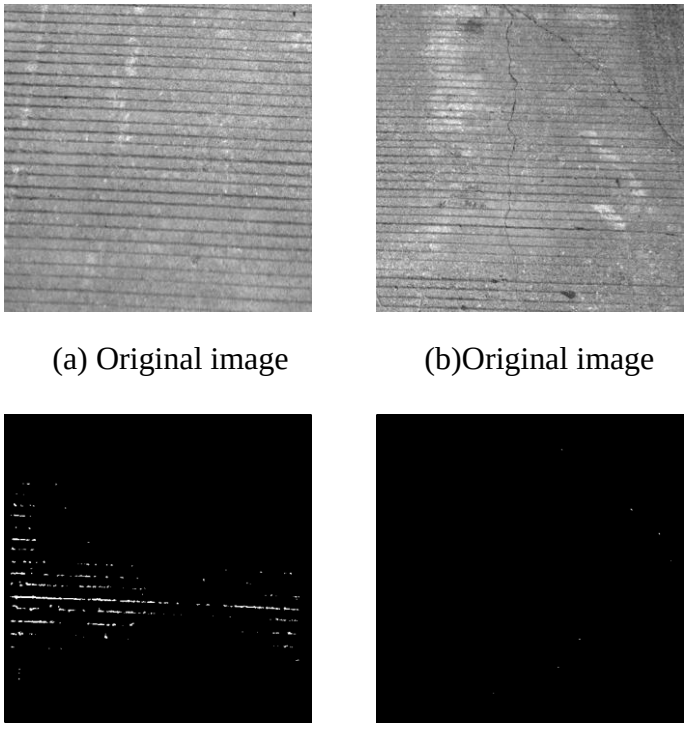
Number of images = 40	Predicted: NO	Predicted: YES
Actual: NO	15	4
Actual: YES	17	2

Table 5. U-Net with pyramid pooling module confusion matrix

Number of images = 40	Predicted: NO	Predicted: YES
Actual: NO	9	11
Actual: YES	7	13

Table 6. Proposed method confusion matrix

Number of images = 40	Predicted: NO	Predicted: YES
Actual: NO	12	8
Actual: YES	6	14



(a) Original image (b)Original image

(c) Output of a (d) Output of b

Figure 9. Example of the misclassified crack image of false positive (left a,c) and false negative (right b,d)

According to the results, U-Net, U-Net with pyramid pooling module, and the proposed model showed an accuracy of 42%, 55%, and 65%, respectively. The result found that U-Net has the majority of crack images predicted as non-crack. This finding

highlights that the module is insufficient in detecting thin cracks. Many output images indicate the U-Net falls into the false negative category in Fig. 9 (right). The second model, U-Net with pyramid pooling module, shows an improved result where almost even test images fall into the four categories in Table 4. However, the network misclassified the horizontal grid as a detected crack, as shown in Fig. 9 (left). The proposed model provides the highest scores out of all. The matrix accuracy is 65%, which is 10% higher compared to U-Net with the pyramid pooling module. These findings demonstrate that the proposed strategy outperforms other methods in thin crack segmentation.

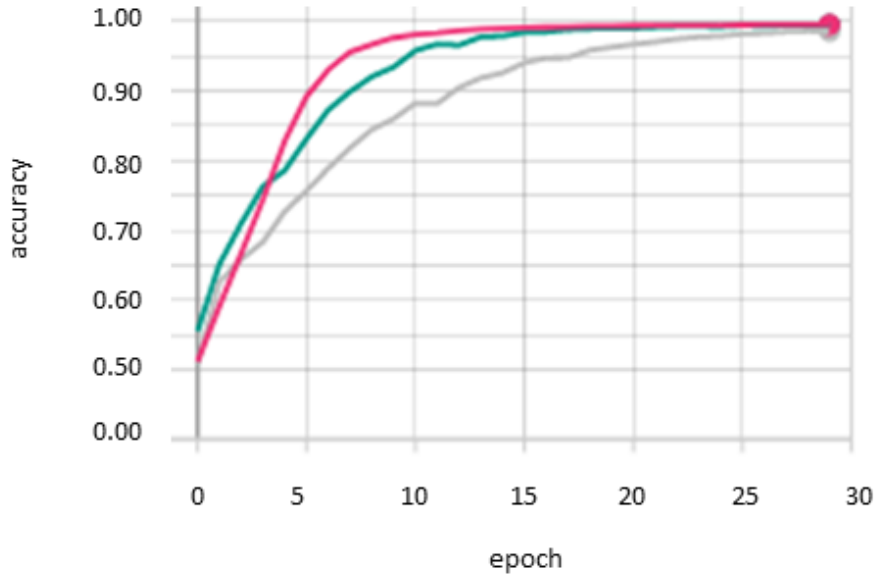


Figure 10. Accuracy of training. Grey represents U-Net, green represents U-Net with pyramid pooling module, red represents the proposed method

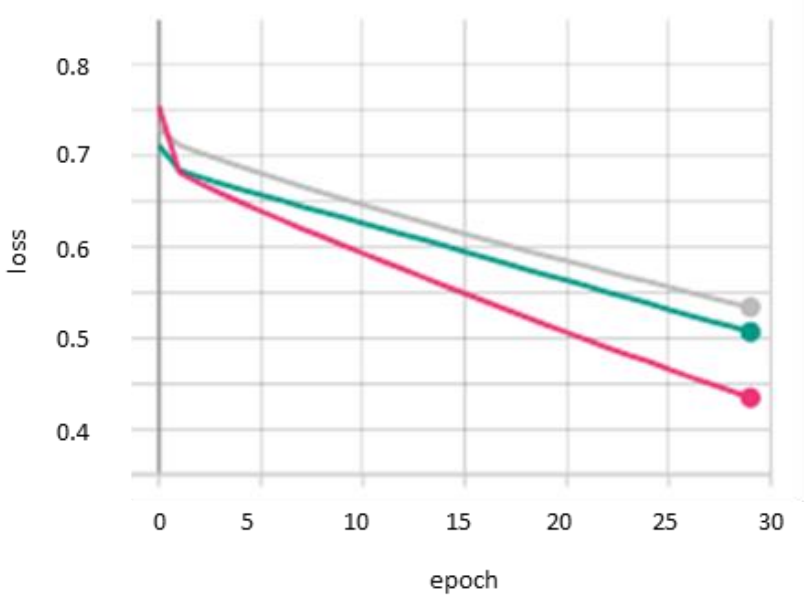


Figure 11. Loss of training. Grey represents U-Net, green represents U-Net with pyramid pooling module, red represents the proposed method

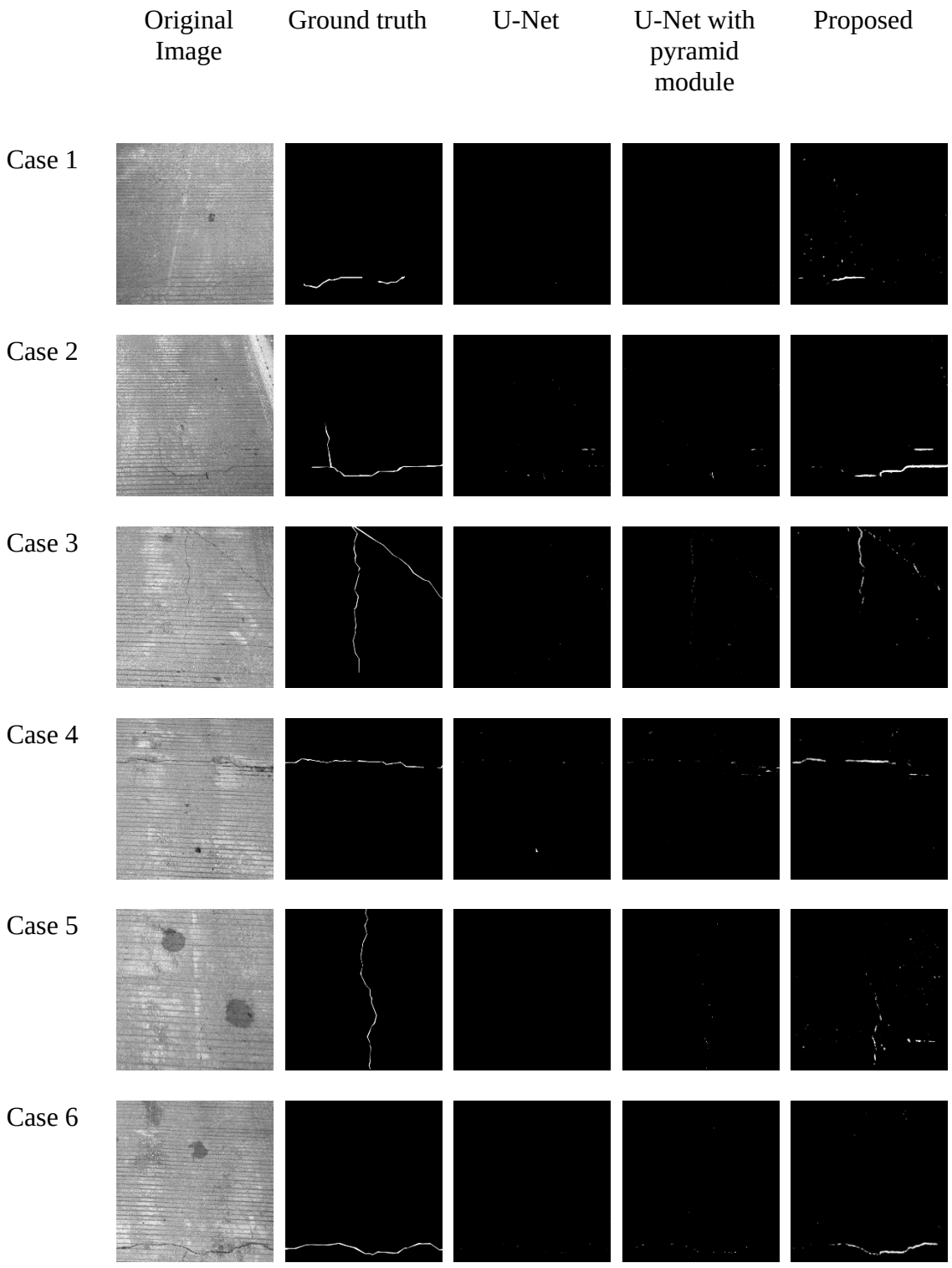


Figure 12. Implementation results

5. CONCLUSION

In the modern era of fast-developing technology, computer vision semantic segmentation is invaluable for concrete crack detection. Edge detection FHT was initially developed but proved to be inadequate due to their susceptibility to misclassification of crack-like noises. Since then, CNNs such as FCNs have been the backbone for crack semantic detection. FCNs override the downside of FHT by labeling each pixel and identifying the cracks into appropriate classes. Unfortunately, manual inspections are still performed despite their inefficiency as a result of high cost and FCNs' inaccuracy to find thin cracks.

In this paper, we modeled an enhanced end-to-end FCN semantic segmentation to identify thin concrete cracks. We chose our model based on U-Net because studies showed it has the highest semantic segmentation accuracy when considering the number of images used in comparison to Cha's CNN and VGG16 [1, 22, 23]. We employed U-Net's symmetric architecture and enhanced it through the concatenation of multiple convolutional layers to reinforce features trained on each down-sampling. Additionally, we employed a pyramid pooling module for an improved scene parsing.

The performance of the proposed model was compared with the original U-Net and U-Net with the pyramid pooling module using the confusion matrix. Since U-Net already showed an excellent performance using the dataset from [10], we developed a custom dataset with robust classification models. This custom dataset had additional ambiguity such as concrete filler, water spots, and protruding horizontal grids generated from the original cast for better detection of thin cracks. To verify the robustness of the method, we ran the proposed method using data that consist of images a half with crack and half without crack. The proposed model indeed had a better detection result. The

model had a 23% and 10% better detection rate compared to U-Net and U-Net with pyramidal pooling module, respectively.

To further the research in the future, we would like to optimize the number of parameters of the proposed architecture. In our experiment, the U-Net, U-Net with pyramid pooling module, and proposed architecture consist of 2,439,361, 37,353,893, and 42,515,853 total parameters, respectively. We would like to find the best-optimized case that consists of the least parameter with similar output results of the proposed model by reducing the number of convolutional layers. In addition, we would like to gather the same type of data where the direction of the camera and the surface is perpendicular instead of angled, like in this research. The new data will help analyze whether or not our proposed model is also effective in a non-angled dataset.

LITERATURE CITED

1. C. Dung, L. Anh. Autonomous concrete crack detection using deep fully convolutional neural network., Autom. Constr., 99 (2019), pp. 52-58, 10.1016/j.autcon.2018.11.028.
2. Z. Liu, Y. Cao. Computer vision-based concrete crack detection using U-net fully convolutional networks, Constr., 104 (2019), pp. 129-139, 10.1016/j.autcon.2019.04.005.
3. W. Cook, P.J. Barr. Observations and trends among collapsed bridges in New York state, J. Perform. Constr. Facil., 31 (4) (2017), p. 04017011, 10.1061/(ASCE)CF.1943-5509.0000996.
4. O. Ronneberger, P. Fischer, T. Brox. U-net: Convolutional networks for biomedical image segmentation, arXiv:1505.04597v1 [cs.CV] 18 May 2015.
5. N. Wang, X. Zhao. Automatic damage detection of historic masonry buildings based on mobile deep learning, Autom. Constr., 103 (2019) 53-66.
6. W. Silva, D. Lucena. Concrete Cracks Detection Based on Deep Learning Image Classification, Proceedings 2018, 2, 489; doi:10.3390/ICEM18-05387.
7. Z. Fan, Y. Wu. Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Nerual Network, arXiv:1505.04597v1[cs.CV].
8. G. Felio. Informing the future: the Canadian infrastructure report card, http://canadianinfrastructure.ca/en/index.html, (October 18, 2019).
9. A. Krizhevsky, G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks, in: F. Pereira, C.J.C. Burges, L.Bottou, K.Q.B.T.-A in N.I.P.S. 25(NIPS 2012) Weinberger (Eds), Int. Conf. Neural Inf. Process. Syst., Curran Associates, Inc., 2012: pp. 1-9.

10. Caglar Firat Ozgenel, Concrete Crack Images for Classification, Mendeley Data, v1, (2018), https://doi.org/10.1007/s112632/5y9wdsg2zt.1.
11. Dumoulin, Vincent, en Francesco Visin. A guide to convolution arithmetic for deep learning. arXiv:1603.07285 [cs, stat], March 2016. arXiv.org, http://arxiv.org/abs/1603.07285.
12. W. Liu, A Rabinovich, A. Berg. ParseNet: Looking Wider to See Better., arXiv:1506.04579v2 [cs.CV] 19 Nov 2015.
13. H. Zhao, J. Shin, X. Qi, X. Wang, J. Jia. Pyramid Scene Parsing Network, IEEE Conference on Computer Vision and Pattern Recognition, arXiv:1612.01105v2 [cs.CV] 27 Apr 2017.
14. J. Kim, H. Lee, S. Hong. Objects Segmentation From High-Resolution Aerial Images Using U-Net With Pyramid Pooling Layers, IEEE Geoscience and Remote Sensing Letters, Vol.16,No.1, January 2019.
15. H. Noh, S. Hong, B. Han. Learning Deconvolution Network for Semantic Segmentation, arXiv:1505.04366v1 [cs.CV] 17 May 2015.
16. J. Long. E. Shelhamer, T. Darrell. Fully convolutional networks for semantic segmentation, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Dec. 2015, pp. 3431-3440.
17. A. Mohan, S. Poobal. Crack Detection Using Image Processing: A critical review and analysis, Alex. Eng. J. 2018, 57, 787-798.
18. J. Yan, A. Downey, A. Cancelli, S. Laflamme. Concrete Crack Detection and Monitoring Using a Capacitive Dense Sensor Array. Sensors 2019, 19, 1843.

19. Y. Liu, J. Yao. DeepCracks: a deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139-153, https://doi.org/10.1016/j.neucom.2019.01.036.
20. B. Spencer, V. Hoskere. Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering 5 (2019) 199-222, https://doi.org/10.1016/j.eng.2018.11.030.
21. H. Kim, M. Shin, S. Sim. Crack and Noncrack Classification from Concrete Surface Images Using Machine Learning Structural Health Monitoring (2018), DOI: 10.1177/1475921718768747.
22. M. Islam, J. Kim. Vision-Based Autonomous Crack Detection of Concrete Structures Using a Fully convolutional Encoder-Decoder Network. Sensors 2019, 19(19), 4251.
23. J. Zhang, C. Lu., J. Wang, L. Wang, X. Yue. Concrete Cracks Detection Based on FCN with Dilated Convolution. Appl. Sci. 2019, 9(13), 2686.
24. S. Ioffe, C. Szegedy. Batch normalization: Acceleerating deep network traning by reduction ginternal covariate shift, in Proc. Int. Conf. Mach. Learn, 2015, pp.448-456.
25. M. Drozdzal, E. Vorontsov. The importance of skip connections in biomedical image segmentation in Deep Learning and Data Labeling for Medical Applications, pages 179-187. Springer, 2016.
26. Z. Zhou, M. Siddiquee. UNet++: A Nested U-Net Architecture for medical Image Segmentation, arXiv:1807.10165v, 2018.

27. G. Huang, Z. Liu. Densely Connected Convolutional Networks, arXiv: 1608.06993v5, 2018.

28. D. Kingma, J. Ba. Adam: a method for stochastic optimization, https://arxiv.org/abs/1412.6980.

29. C. Szegedy, V. Vanhoucke. Rethinking the inception architecture for computer vision, In Proceedings of the IEEE conference on computer vision and pattern reconition, pages 2818-2826, 2016.

30. S. Dorafshan. R. Thomas, M. Maguire. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete, Constr. Build. Mater. 186(2018) 1031-1045, https://doi.org/10.1016/j/conbuildmat.2018.08.011.

31. T. Yamaguchi, S. Hashimoto. Fast crack detection method for large-size concrete surface images using percolation-based image processing, Mach Vis. Appl. 21 (2010) 797-809, https://doi.org/10.1007/s00138-009-0189-8.

32. B. Liu, T. Yang. Image analysis for detection of bugholes on concrete surface, Constr. Build. Mater. 137 (2017) 432-440, https://doi.org/10.1016/j.conbuildmat.2017.01.098.

33. H. Nhat-Duc, Q. Nguyen, V. Tran. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network, Autom. Constr. 94 (2018) 203-213, https://doi.org/10.1016/j.autcon.2018.07.008.

34. W. Liu, A. Rabinovich, A. C. Berg. Parsenet: Looking wider to see better. arXiv:1506.04579, 2015.

35. K. Gopalakrishanan, S. Khaitan, A. Choudhary, A. Agrawal. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. Construction and Building Materials 157(2017) 322-330.

36. Y. Cha, W. Choi, O. Buyukozturk, Deep learning-based crack damage detection using convolutional neural networks, Comput. Aided Civ. Inf. Eng. 32(2017) 361-378, https://doi.org/10.1111/mice.12263.